

Volume Exploration using Ellipsoidal Gaussian Transfer Functions

Yunhai Wang ^{*1,2}, Wei Chen ^{†3}, Guihua Shan ^{‡1,2}, Tingxin Dong ^{§1,2}, and Xuebin Chi ^{¶1,2}

¹Graduate University of Chinese Academy of Science

²Computer Network Information Center, Chinese Academy of Sciences

³State Key Lab of CAD&CG, Zhejiang University

ABSTRACT

This paper presents an interactive transfer function design tool based on ellipsoidal Gaussian transfer functions (ETFs). Our approach explores volumetric features in the statistical space by modeling the space using the Gaussian mixture model (GMM) with a small number of Gaussians to maximize the likelihood of feature separation. Instant visual feedback is possible by mapping these Gaussians to ETFs and analytically integrating these ETFs in the context of the pre-integrated volume rendering process. A suite of intuitive control widgets is designed to offer automatic transfer function generation and flexible manipulations, allowing an inexperienced user to easily explore undiscovered features with several simple interactions. Our GPU implementation demonstrates interactive performance and plausible scalability which compare favorably with existing solutions. The effectiveness of our approach has been verified on several datasets.

Index Terms: I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism

1 INTRODUCTION

Volume exploration focuses on revealing hidden structures in volumetric datasets. Effective exploration is a challenging problem because there is no prior information available with regard to the data distribution. This difficulty is magnified by the fact that exploring and manipulating in the 3D space is typically counterintuitive and laborious. Transfer function design is a user-controllable process that structures the statistical space (i.e., the histograms) and maps the selected data properties to specific colors and opacities. To better disambiguate the various materials and structures, a number of multi-dimensional transfer functions have been proposed. In particular, the 2D transfer functions [9] based on scalar values and gradient magnitudes are very effective in extracting multiple materials and their boundaries. The specification of 2D transfer functions can be performed by means of various classification widgets. However, the selection of features within the multi-dimensional transfer function is a time-consuming and trial-and-error process, and it is very likely to yield unsatisfactory results. The gap between the flexibility of multi-dimensional transfer function design and the fidelity requirement of the volume exploration makes the transfer function design challenging. This can be witnessed by the huge amount of efforts being devoted to the development of intuitive transfer function schemes.

We have identified three reasons for this. First, the search space for finding transfer functions is very large in high-dimensional feature spaces. The user is often required to spend a great deal of time exploring to understand the underlying features and their spatial relationships within the data set. Second, traditional classification widgets are not easily used to represent certain features located within multi-dimensional spaces which have complex shapes because most of them are axis-aligned. Third, modulating the parameters of the classification widgets to maximize the likelihood of feature separation is not trivial, even when all features have been identified.

This paper presents a novel volume exploration scheme based on the argument that Gaussian transfer functions are well suited for classification with a unique combination of multiple data values that localize features in the transfer function domain [10]. Our new scheme takes an analyze-and-manipulate routine, where prior to manipulation, it performs a maximum likelihood feature separation of the statistical space to construct a continuous and probabilistic representation with a Gaussian Mixture Model (GMM). The GMM provides an automatic and suggestive volume classification by converting its mixture components to a set of ellipsoid Gaussian transfer functions (ETFs). This also leads to a novel transfer function design tool in the context of the Gaussian transfer function scheme [10]. Manipulating the shape, size, orientation, associated color and opacity of the ETFs allows for flexible volume exploration while preserving the characteristics of the underlying volume space. Furthermore, the GMM and associated ETFs can be iteratively altered by the user to maintain the properties of feature separation and ease the volume exploration. Figure 1 shows a two-step exploration procedure.

Unlike previous methods [15, 21] which simply display, merge or split the clusters, we provide a suite of intuitive user widgets to ease the interactions thanks to the flexibility of the Gaussian functions. For each Gaussian, the user is capable of performing translation, rotation, scaling and division operations. Similar to traditional transfer function widgets, the user can freely manipulate some widgets to locate interesting features. For instance, rotating the Gaussian widgets helps the user find interesting features in different directions, while scaling can expand and shrink the regions within the feature space. To find small-scale features in large-scale Gaussians, our interface allows the user to subdivide it into smaller ones. Possible operations include direct subdivision of existing ones and insertion of new mixture components into the existing mixture by using the greedy Expectation-Maximization algorithm. Using arbitrary directional Gaussian transfer functions facilitates the representation of features within a complex shape, and can be integrated into the pre-integration volume rendering process based on the analytic integral formation [10].

The rest of this paper is organized as follows. The related work is summarized in Section 2. The maximized likelihood volume classification is described in Section 3, followed by Section 4 that introduces an interactive classification interface. We

*e-mail: wyh@sccas.cn

†e-mail:chenwei@cad.zju.edu.cn

‡e-mail:sgh@sccas.cn

§e-mail:txdong@sccas.cn

¶e-mail:chi@sccas.cn

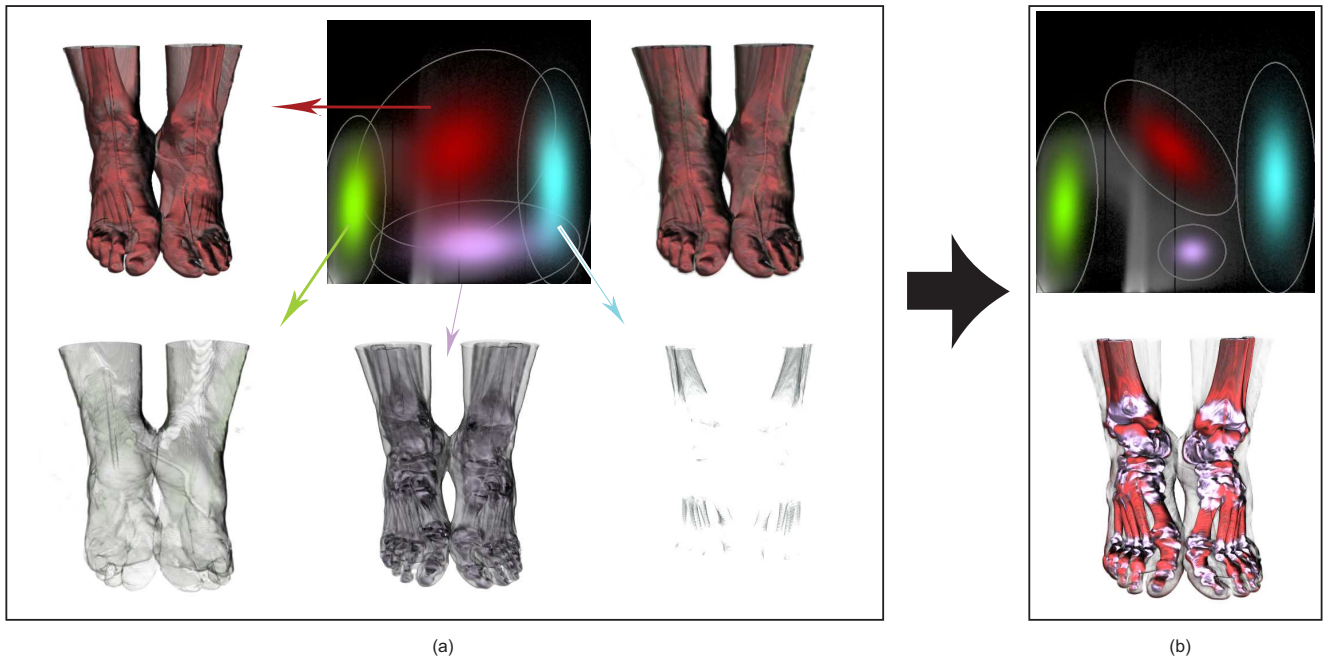


Figure 1: Modeling the (density, density gradient magnitude) 2D histogram with the GMM for the Feet dataset. (a) The automatically generated Gaussian transfer functions, the corresponding volume rendering, and volume renderings associated with each mixture component. (b) The result obtained by scaling the red and plum Gaussian transfer functions and adjusting their maximum opacities, in which the relationship between the toes and ankles is clearly shown.

present the implementation details in Section 5, and demonstrate the results in Section 6. Finally, we conclude the paper in Section 7.

2 RELATED WORK

Multi-Dimensional Transfer Functions A complete review of existing transfer function design is beyond the scope of this article. We refer the reader to Pfister et al. [14]. We restrict our discussion to the design of multi-dimensional transfer functions [11]. In spite of its excellent performance in materials classification, multi-dimensional transfer functions have not received widespread attention until the important work by Kindlmann and Durkin [7]. Their research showed that determining multi-dimensional transfer functions in a 2D histogram of data values and gradient magnitudes can effectively capture boundaries between materials. To facilitate the multi-dimensional transfer function specification, Kniss et al. [9] introduced a set of manipulable widgets. Local [8, 16] and global information [3, 4] can be incorporated into the multi-dimensional transfer function domains. However, determining appropriate transfer functions with traditional classification widgets is a trial-and-error process, particularly because conventional transfer function widgets lack the necessary information needed to guide the user towards a correct feature separation.

Intuitive Transfer Functions Design There has been a great deal of excellent work done to simplify the creation of multi-dimensional transfer functions. Tzeng and Ma [21] used the ISO-DATA algorithm to perform clustering in the multi-dimensional histograms. Roettger et al. [15] proposed spatial transfer functions that consider spatial information in the process of clustering 2D histograms. To effectively structure the feature space, Selver and Güzelis [17] use a self-generating hierarchical radial basis function network to analyze the volume histogram stacks and Maciejewski et al. [12] applied a non-parametric density estimation technique. To make transfer function design as easy as painting, Tzeng et al [20] introduced a painting interface that derives high-dimensional transfer functions by using machine learning techniques. In this paper, we introduce a new volume exploration scheme by analyzing the

histogram space with the Gaussian mixture model, maximizing the likelihood of feature separation. Instant visual feedback is possible by mapping these Gaussians to ellipsoid Gaussian transfer functions (ETFs) and analytically integrating these ETFs in the context of the pre-integrated volume rendering process.

Gaussian Mixture Model The Gaussian Mixture Model (GMM) [1] is well-suited to the modeling of clusters of points. In this model each cluster is assigned a Gaussian, with its mean somewhere in the middle of the cluster, and with a standard deviation that measures the spread of that cluster. It has been widely used in pattern recognition [19]. GMM has also been used in the visualization community. For instance, Correa et al. [2] used it to model uncertainty distributions.

By removing the normalized term from the GMM, the mixture components become a sum of the Gaussians. This is exactly the Gaussian transfer functions used in multi-dimensional transfer function design [10, 18]. Note that, most of these methods utilize axis-aligned ellipsoidal Gaussian transfer functions. In contrast, in our approach the mixture components of the GMM are represented as a sum of arbitrary directional Gaussian transfer functions, which have been proven to be very efficient in representing complex shapes [6].

3 EXPLORING THE FEATURE SPACE WITH THE GMM

GMM is an unsupervised process that yields a continuous and probabilistic feature representation, and can be used to extract coherent regions in the feature space and corresponding coherent structures in the input data space [1]. We choose to use the GMM to explore the 2D histogram space for two reasons. First, the parameters of the GMM are determined by means of the maximum likelihood principle and the Expectation-Maximization algorithm, effectively characterizing the feature space. With that, the conventional 2D transfer function editing can be more convincing and reasonable. Second, the user can flexibly manipulate them to explore unknown features by mapping mixture components to the Gaussian transfer functions. Due to the integration of a Gaussian function being ana-

lytic, Gaussian transfer functions can be effectively pre-integrated, and evaluated on commodity graphics hardware.

3.1 Maximized Likelihood Feature Separation

With a 2D histogram space (e.g., the density vs the gradient magnitude), the semi-parametric Gaussian mixture model can be used to estimate the distribution based on the assumption that the histogram distribution is generated by a mixture of Gaussian functions. Each homogenous region in the feature space is represented by a Gaussian distribution:

$$g(x|\mu, \Sigma) = \frac{1}{2\pi|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (1)$$

where μ is the mean value and Σ is a 2×2 covariance matrix in the 2-D space. The distribution of all regions is represented by a Gaussian mixture model:

$$p(x|\theta) = \sum_{j=1}^k \alpha_j g_j(x|\mu_j, \Sigma_j) \quad (2)$$

where α_j is the mixing probability of the j th Gaussian where

$$\sum_{j=1}^k \alpha_j = 1 \quad \text{and for } j \in \{1, \dots, k\} : \alpha_j \geq 0$$

θ denotes the parameter set of the GMM with k components:

$$\theta = \{\alpha_1, \dots, \alpha_k, \mu_1, \dots, \mu_k, \Sigma_1, \dots, \Sigma_k\} \quad (3)$$

Because the set of mixtures of Gaussian functions is parametric, the density estimation problem can be defined more specifically as a problem of finding the set of parameters θ that specifies the model to which the points in the feature space most likely belong. Given a set of feature vectors x_1, \dots, x_n , the maximum likelihood estimation of θ is:

$$\hat{\theta} = \arg \max_{\theta} p(x_1, \dots, x_n | \theta) = \arg \max_{\theta} \prod_{i=1}^n p(x_i | \theta) \quad (4)$$

The Expectation-Maximization (EM) algorithm [5] provides an iterative means to determine θ . Given an initial estimated parameter set θ , the EM algorithm iterates over the following steps until it converges to a local maximum of the likelihood function:

• E Step:

$$P(j|x_i) = \frac{\alpha_j p(x_i|\mu_j, \Sigma_j)}{\sum_{j=1}^k \alpha_j p(x_i|\mu_j, \Sigma_j)} \quad i = 1, \dots, n, \quad j = 1, \dots, k \quad (5)$$

• M Step:

$$\hat{\alpha}_j = \frac{1}{n} \sum_{i=1}^n P(j|x_i) \quad \hat{\mu}_j = \frac{\sum_{i=1}^n P(j|x_i) x_i}{\sum_{i=1}^n P(j|x_i)} \quad (6)$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^n P(j|x_i) (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T}{\sum_{i=1}^n P(j|x_i)}$$

The E step finds the expected value of the log-likelihood $\sum_{i=1}^n \log p(x_i|\theta)$ and the M step finds new parameters to maximize the expectation computed in the E step. With a few iterations, this algorithm converges to a locally optimal solution. However, the convergence to a globally optimal solution is not guaranteed and the number of iterations depends on the initial assigned parameters. As for some complicated data, this dependency requires the user to spend a great deal of time to find properly initialized parameters. For most practical data, the optimal number of mixture components

k is unknown, this requires that the user try many different numbers. This poses a difficulty for interactive volume classification.

To overcome these drawbacks, the greedy EM algorithm [22] builds the model in an adaptive manner. After starting with a single component whose parameters are trivially computed, two steps are alternatively performed: adding a new component to the mixture, and updating the complete mixture using the above mentioned E step and M step until certain criteria are met. With this greedy algorithm, the initial parameters θ do not need to be assigned by the user, and the number of mixture components becomes manageable. If the results are not satisfying, the user can insert new components to update the GMM. Fig. 2 shows the volume classification results of the Engine dataset using this greedy algorithm. In Fig. 2 (b), a new component is inserted to the clustering result shown in Fig. 2 (a), classifying the main body and wheels parts.

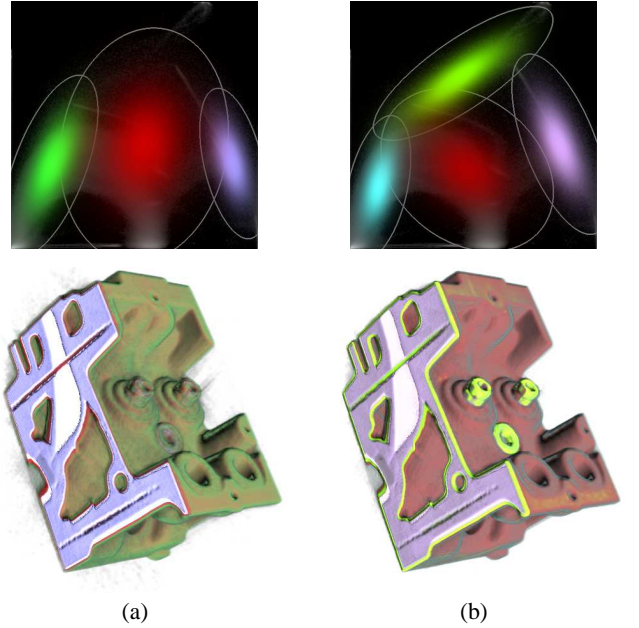


Figure 2: Using the greedy EM algorithm to classify the Engine data with different number of mixture components: (a) 3; (b) 4. In (b), the main body and wheel parts are classified.

After finding an appropriate θ , each pixel in the feature space is associated with a probability vector $p = (p_1, \dots, p_k)$ such that $p_i = g(x|\theta_i)$. With these vectors, the discrete 2D histogram space becomes a continuous space. Different from the non-parametric kernel density estimation [12], the GMM is a semi-parametric density estimation technique, where an analytical Gaussian function is used to represent each cluster. This property greatly favors the clustering results with a set of transformations as described later.

3.2 Ellipsoid Gaussian Transfer Functions

One important advantage of the GMM based separation over previous work [15, 21] is that the obtained mixture components can be converted to ellipsoidal Gaussian transfer functions (ETF)

$$\rho(x) = \alpha_{\max} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (7)$$

where α_{\max} is the maximum opacity of the ETF and the symmetric covariance matrix provided by EM algorithm is defined as:

$$\Sigma^{-1} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad (8)$$

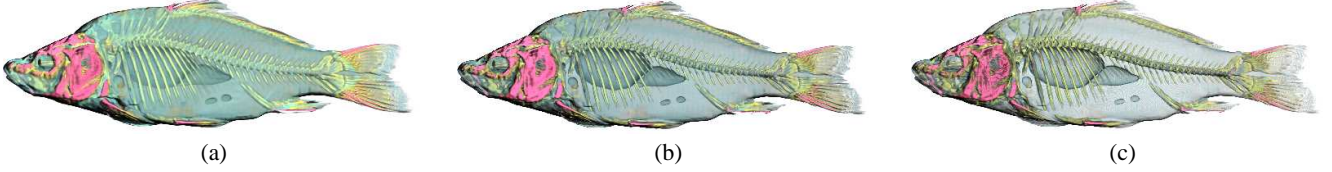


Figure 3: Visualizing the Carp dataset ($256 \times 256 \times 512$) with 3 ETFs by (a) analytic integration at the object sample distance 0.6; (b) numerical integration at the object sample distance 0.3; (c) numerical integration at the object sample distance 0.6. Their performances are 23fps, 13fps, 38fps, respectively.

where b can be non-zero. Compared to the axis-aligned GTF used in [10, 18] where Σ^{-1} is a diagonal matrix, our ETF is more general and affords more flexible feature separation. Since the mixing probability of each Gaussian represent its maximal contribution to density distribution, we can use it to set initial α_{\max} for each ETF. **Automatic Color Assignment** According to the principles of color design [23], the luminance is important for perception of transparency and the hue plays a major role for visual labeling. Thus, we propose a perception-enhanced coloring scheme, which automatically assigns an HSL-based color to the j th ETF:

$$h_j = \frac{j \times 360}{k}, \quad s_j = 1.0, \quad l_j = \frac{j+m}{k+m} \quad (9)$$

where m is an integer between 0 and k to avoid dummy luminance.

3.3 Pre-integrated Volume Rendering with the ETFs

Kniss et al. [10] and Song et al. [18] have derived analytic forms to pre-integrate axis-aligned Gaussian transfer functions. In this section, we demonstrate that an arbitrarily directional ETF can also be incorporated with the pre-integrated volume rendering.

According to the volume rendering equation [13], the opacity can be expressed as:

$$\begin{aligned} \alpha &= 1 - \exp\left(-\int_0^D \alpha(\mathbf{x}(\lambda)) d\lambda\right) = 1 - \exp\left(-\int_0^D \sum_{j=1}^n \rho_j(\mathbf{x}(\lambda)) d\lambda\right) \\ &= 1 - \exp\left(\sum_{j=1}^n \left(-\int_0^D \rho_j(\mathbf{x}(\lambda)) d\lambda\right)\right) \end{aligned} \quad (10)$$

where D is the distance between the entry and existing sampling points f and b . By assuming that the feature vector x between \mathbf{x}_f and \mathbf{x}_b vary linearly, the integration term $-\int_0^D \rho_j(\mathbf{x}(\lambda)) d\lambda$ in Equation 10 becomes:

$$\begin{aligned} R_j &= -\int_0^D \rho_j(\mathbf{x}(\lambda)) d\lambda \\ &= -\int_0^D \alpha_{\max j} e^{-\frac{1}{2}(\mathbf{x}_f - \mu_j + \lambda \frac{\mathbf{x}_b - \mathbf{x}_f}{D})^T \Sigma_j^{-1} (\mathbf{x}_f - \mu_j + \lambda \frac{\mathbf{x}_b - \mathbf{x}_f}{D})} d\lambda \end{aligned} \quad (11)$$

Suppose the feature vector \mathbf{x} consists of a density component and a gradient magnitude component, we have $\mathbf{x} = \{s, g\}$, $\mu_j = \{s_j, g_j\}$, $\mathbf{x}_f = (s_f, g_f)$, and $\mathbf{x}_b = (s_b, g_b)$. We define $k_s = \frac{s_b - s_f}{D}$, $k_g = \frac{g_b - g_f}{D}$, $d_s = s_f - s_j$, $d_g = g_f - g_j$, yielding:

$$R_j = -\int_0^D a_{\max} e^{-\frac{1}{2}(a(k_s \lambda + d_s)^2 + 2b(k_s \lambda + d_s)(k_g \lambda + d_g) + c(k_g \lambda + d_g)^2)} d\lambda$$

Let $A = \sqrt{ak_s^2 + 2bk_s k_g + ck_g^2}$, $B = (ak_s d_s + b(k_s d_s + k_g d_g) + ck_g d_g) / A$, $C = (ad_s^2 + 2bd_s d_g + cd_g^2) - B^2$, R can be written as:

$$\begin{aligned} R_j &= -\int_0^D a_{\max} e^{-\frac{1}{2}((A\lambda + B)^2 + C)} d\lambda \\ &= -a_{\max} \sqrt{\frac{\pi}{2}} \frac{e^{-\frac{C}{2}}}{A} \left(\operatorname{erf}\left(\frac{AD+B}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{B}{\sqrt{2}}\right) \right) \\ &= -P \cdot \left(\operatorname{erf}\left(\frac{AD+B}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{B}{\sqrt{2}}\right) \right) \end{aligned} \quad (12)$$

where $\operatorname{erf}(z) = \int_0^z e^{-x^2} dx$ and $P = a_{\max} \sqrt{\frac{\pi}{2}} \frac{e^{-\frac{C}{2}}}{A}$. When A is less than or equal to zero, R_j can be evaluated with Equation 7. The final opacity is $\alpha = 1 - \exp(\sum_{j=1}^n R_j)$.

Previous work [10, 18] approximates the erf function with a 2D texture, and thus is a means of numerical integration. Instead, we analytically evaluate the form by leveraging the newest GPU, leading to high quality pre-integrated volume rendering. Fig. 3 (a,c) compares our result with that of numerically integrated rendering at the same sampling rate. Fig. 3 (b) shows the result from the numerical integration scheme when performed at a doubled sampling rate. Although comparable visual effects are achieved in Fig. 3 (a,b), our approach is much faster than the numerical integration approach.

4 OUR EXPLORATION WIDGETS

With an input dataset and its associated histogram, our approach employs EM clustering to automatically compute a set of suggestive ETFs at the beginning. The choice of appropriate number of components in clustering can be achieved in a greedy manner as mentioned in section 3.1, rather than determining them in the first place [21, 12]. This could be very helpful for inexperienced users. Surely, if the user has adequate knowledge about the data, we also allow the user to define new ETFs to explore the data without utilizing the GMM clustering. This is the major difference between our volume exploration and previous work [15, 21], where the user's operations are limited in the clustered space. If the initial ETFs defined by the user or provided by the EM clustering are not satisfying, the user has to change the ETFs by using something other than a recoloring operation. As shown in Fig. 1 (a), the features determined by the red ETF are very large, but by solely adjusting its α_{\max} , it is difficult to show other clear structures. Accordingly, we designed a suite of exploration widgets to allow the user to freely manipulate the ETFs. Fig. 4 illustrates the pipeline of our approach.

4.1 Ellipsoid Classification Widgets

Unlike the inverse triangular or rectangular widgets used in previous work [10, 18], the manipulation primitives in our approach are arbitrarily directional ellipsoidal ETFs, and the operations can be represented as a variety of transformations to the covariance matrix Σ . The center of the ellipse is the mean value μ , and other parameters can be computed by applying the singular value decomposition algorithm to the matrix Σ^{-1} :

$$\begin{aligned} \Sigma^{-1} &= \begin{bmatrix} a & b \\ b & c \end{bmatrix} \\ &= \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \sigma_0 & 0 \\ 0 & \sigma_1 \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}^T \end{aligned} \quad (13)$$

where $\frac{1}{\sqrt{\sigma_0}}$ and $\frac{1}{\sqrt{\sigma_1}}$ are the ellipse radii along the major and minor axes, ϕ is the angle between the x -axis and the major axis of the ellipse, and ψ is the angle between the minor-axis and the x -axis.

For a symmetry 2×2 matrix, ψ is equal to ϕ . The following affine transformations can be applied:

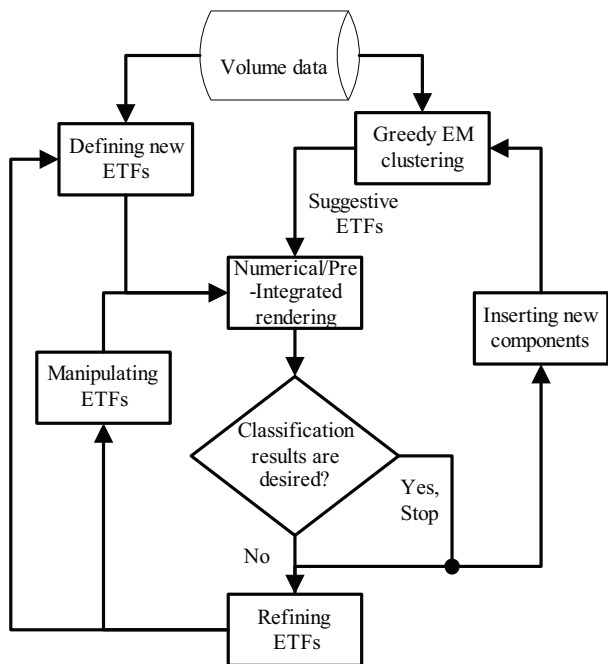


Figure 4: The pipeline of our approach.

- **Translation**- shifting the mean μ . The user can move the widget in the histogram space to explore interesting features.
- **Rotation**- rotating the covariance matrix Σ . This adds an angle β to the ϕ and ψ and substitutes in a new ϕ and ψ in Equation 13, leading to a new covariance matrix Σ .
- **Scaling**- scaling the radii of the principle axes σ_0 and σ_1 , enlarging or shrinking the Gaussian along its tiled angle ϕ .

Fig. 5 demonstrates four operations to the red ETF shown in Fig. 1 (a): recoloring, translating, scaling and rotating. By interactively specifying each mixture component, the corresponding feature regions can be observed, providing a context to modulate the transfer functions. Although there are no guidance in these transformations, the better classification results can be easily achieved.

- **Subdivision** Often, some mixture components may include more than one feature, as illustrated by the dark red ETF in Fig. 2 (a). To find more interesting small-scale features in an ETF, two operations can be performed. First, the greedy EM algorithm can be employed again to yield more ETFs (see Fig. 2). This is done automatically. Meanwhile, the transformations done by the user cannot be incorporated into the subdivision. This is also a limitation of the current EM algorithm. An alternative solution is to directly subdivide an ETF into two pieces along the major axis, yielding two half radii for σ_0 and σ_1 . Further, the user can use the scaling operations to refine them when this subdivision obscures some interesting features. Fig. 6 shows an example, where the joints of the feet can be distinguished from other structures with the subdivision widget. Obviously, we cannot expect the subdivision always lead to a better classification for data features. If the classification results are unsatisfactory, the user can easily backtrack to the original ETF or delete the unwanted one and refine them toward better classification.

5 IMPLEMENTATION DETAILS

We have implemented and tested our system on a PC with a Core 2 Duo E6320 1.8 GHZ CPU, 2.0 GB RAM and an NVIDIA Geforce 8800 GTS video card (256 MB video memory) using Cg Language. All images shown in this paper were generated at a resolution of 1024×768 . The implementation includes two parts: the greedy EM

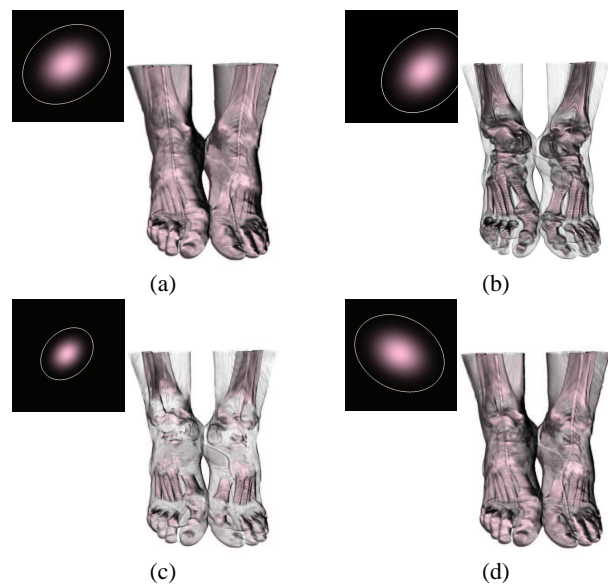


Figure 5: Manipulating the red ETF shown in Fig. 1 (a). (a) Recoloring. (b) Translating the recolored ETF in (a) 0.21 along the x -axis and -0.15 along the y -axis. (c) Scaling the recolored ETF in (a) 0.38 along both the major and minor axes. (d) Rotating the recolored ETF in (a) 45 degrees counter-clockwise.

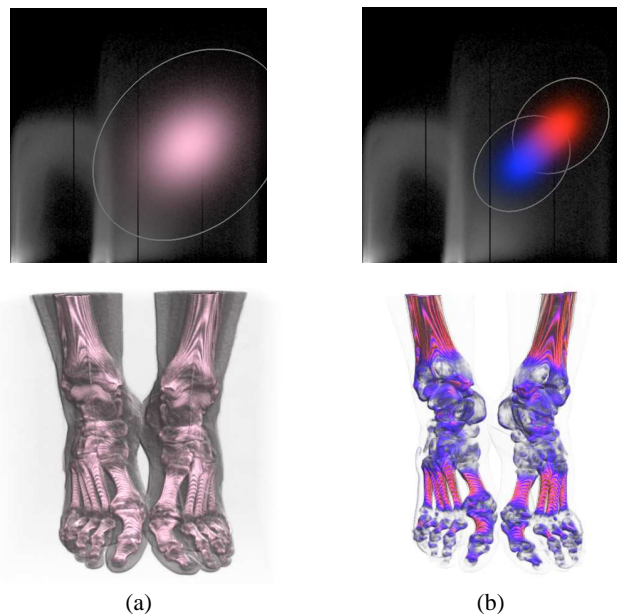


Figure 6: Subdividing an ETF into two smaller ones. (a) An ETF and the rendering result. (b) The recolored and subdivided two ETFs and associated results, where the joints are differentiated from other structures.

algorithm and ray casting based pre-integrated volume rendering. We use the accelerated greedy EM algorithm [22] which has proven to be convergent for large data sets. It first partitions the data into blocks with kd -tree to pre-compute the statistical variables used in the optimization, and then uses the partitioned blocks to perform the optimization.

There are two fashions of using the greedy EM clustering algorithm. The first one performs the EM clustering only once in the entire procedure, i.e., a maximum number of components are generated in the first stage. The components can be freely merged and

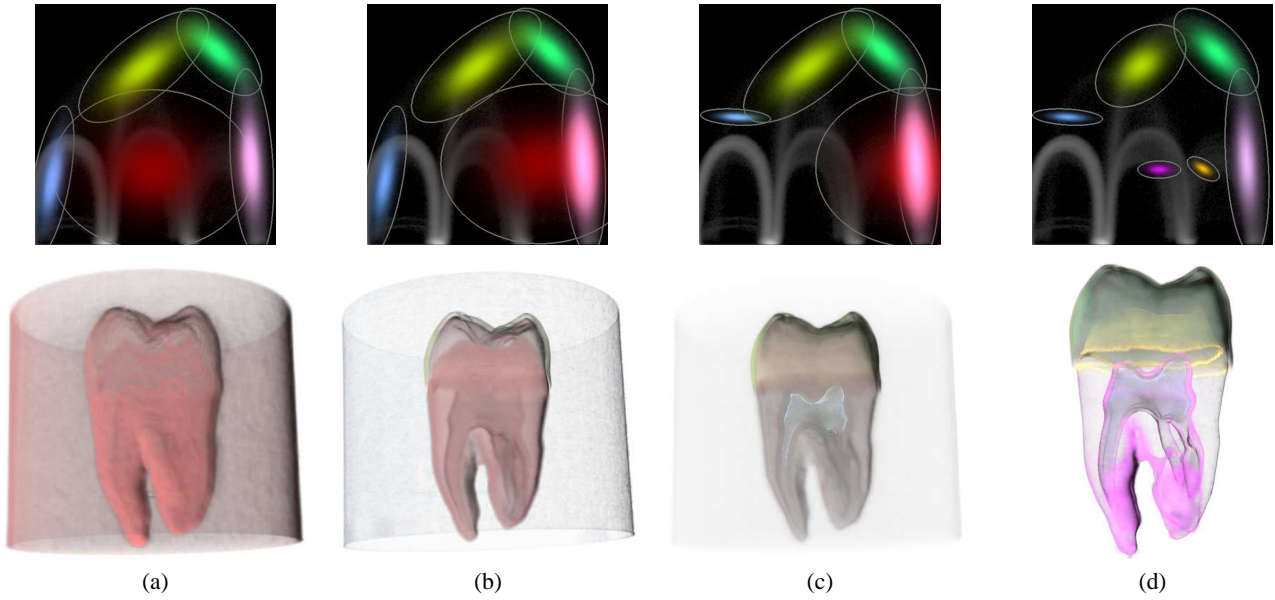


Figure 7: Progressive exploration of the Teeth ($256 \times 256 \times 161$) dataset using the (density, density gradient magnitude) histogram. (a) The automatically generated ETFs and rendering results. (b) Distracting regions are removed when the dark red ETF is deleted, yielding clearer visualization of the pulp and the dentin. (c) The user moves, scales and rotates the blue ETF to discover the pulp. (d) The user scales and subdivides the dark red ETF, and scales the green ETF. All structures are clearly shown.

Nr	One-round	Multi-round	Numerical	Analytic
3	413ms	413ms	22 fps	19 fps
4	795ms	262ms	16 fps	12 fps
5	1301ms	273ms	14 fps	9 fps
6	2289ms	293ms	11 fps	7 fps

Table 1: The component number (the first column), the clustering performance (the second and third columns) and the rendering performance (the fourth and fifth columns) with numerical integration and analytic integration for the Engine dataset ($256 \times 256 \times 128$). The image resolution is 1024×768 .

manipulated, but not added. We call it the *one-round* clustering. In contrast, the second fashion takes a *multi-round* procedure, i.e., new components can be inserted progressively based on an arbitrary initial separation. The later is preferred in our system because it achieves both higher performance and greater flexibility than the one-round scheme. The second and third columns of Table 1 compare the performance of these two fashions.

Evaluating the analytic form of the ETFs for pre-integrated volume rendering on the GPU induces some computational overhead, as demonstrated by the comparison to conventional lookup table enabled numerical integration (see the fourth and fifth columns of Table 1). However, our analytic integration mode can yield results of comparable quality at a lower sampling rate in ray casting than that produced by the numerical integration. Note that the rendering performance gradually decreases with the increasing number of components which is listed in the first column of Table 1.

In our experiments, we found that a small number of mixture components is sufficient for a good approximation to the distribution of the 2D feature space meanwhile offering a rich space of exploration. As shown in Table 1, the generation of 5 mixture components typically requires 1.3 seconds with an unoptimized CPU implementation.

6 APPLICATIONS

The statistical properties contained in the 2D histogram can be arbitrary and data-dependent. For conventional CT or MRI datasets, the density and density gradient magnitude are two commonly used features. By applying other derived properties like the second-order differential properties, or the multi-variables of the dataset itself, the user is equipped with a exploration tool for free feature exploration, transfer function design, data comparison, as well as knowledge assisted multivariate volume visualization.

6.1 Feature Space Exploration

Kindlmann and Durkin [7] noticed that materials' boundaries appear as arcs in the (density, density gradient magnitude) space. The user can search for the boundaries by selecting peaks of the arches [9]. Our approach explores the feature space from another viewpoint based on the well established statistical and probabilistic theories. One distinctive advantage of our approach is that a set of suggestive ETFs is automatically generated, which simplifies the user interaction, and reduces the user's workload. Still, the suggestive ETFs can be iteratively updated, leading the user to the real spatial relationship within the underlying dataset. The efficiency of this scheme has been demonstrated on a variety of datasets including anatomical and flow simulation datasets.

Fig. 1 shows an exploration process for the Feet dataset. The first step, i.e., the automatic classification by means of the maximized likelihood algorithm, produces four ETFs, of which the red and plum ETFs dominate. The red one corresponds to the skin and bone parts, and the plum determines parts of the skin and the ankle part. Most of the voxels in the dataset belong to these two ETFs. However, the skin by the red ETF hides the phalanges and the ankle, and parts of the skin by the plum ETF occludes the main ankle parts. Scaling these two ETFs and adjusting their maximum opacities can clearly differentiate the phalange and ankle. Fig. 1 (b) shows the adjusted results.

In the above examples, all of the important structures are identified with the automatically generated ETFs. For many other datasets, however, most parts of the obtained clusters are noise

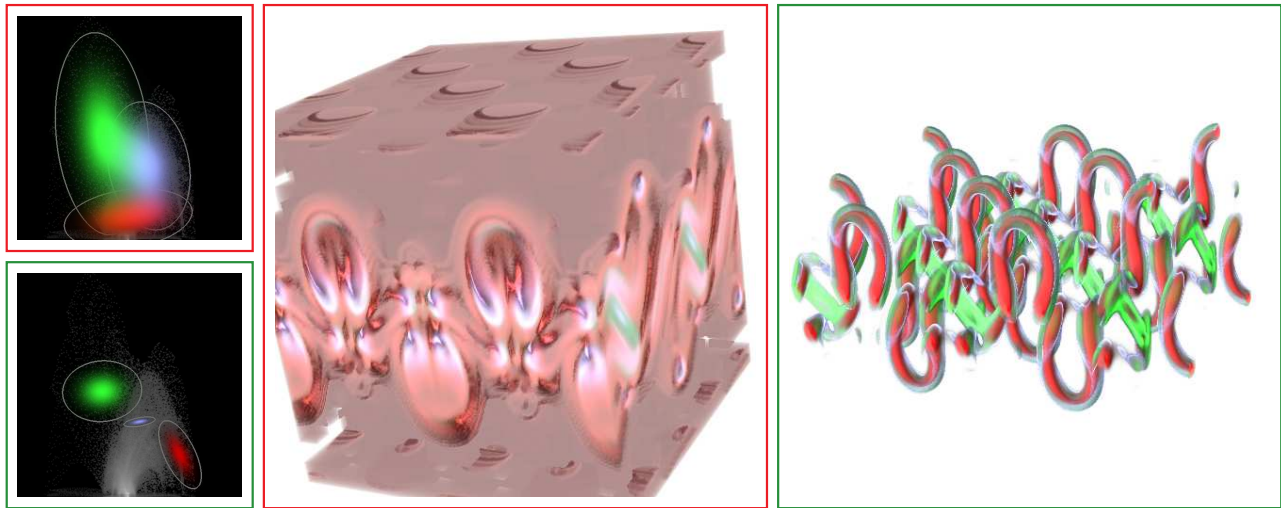


Figure 8: Exploring the Horseshoe Vortex dataset using the (the second invariant of the velocity gradient, its gradient magnitude) feature space. The automatically generated 3 ETFs (top left) produces a result (middle) that shows some noise in the the vortex region. By moving the plum ETF and scaling three ETFs (bottom left), the two layered vortex tubes are clearly shown (right).

while only a small region corresponds to interesting structure. The main reason is that the maximized likelihood algorithm assumes a probability distribution for the entire dataset, and it therefore can not get rid of noisy structures generated by the ETFs. One such example is the dark red ETF (see Fig. 7 (a)) that yields a large amount of noise and very little meaningful information. Translating this ETF can improve the results, as shown in Fig. 7 (b). Further, the user may observe that the steel blue ETF contains some parts of the pulp, and some noise in the region of low gradients. By translating it to the region of higher gradient magnitudes, scaling and rotating it, the pulp is revealed (see Fig. 7 (c)). As a result of the dark red ETF containing the dentin and some parts of the pulp, not all of the structures are visible even after the maximum opacity of the dark red ETF is lowered. This problem can be addressed by scaling and subdividing the dark red ETF into two parts, as shown in Fig. 7 (d).

Fig. 8 shows a two-stage exploration of the Horseshoe Vortex dataset using the (the second invariant of the velocity gradient, its gradient magnitude) feature space. Notice that some noises induced by the medium purple and plum ETFs (see the middle one in Fig. 8). After moving the plum ETF and scaling three ETFs, two layered vortex tubes are clearly shown (see the right one in Fig. 8).

6.2 Knowledge Assisted Visualization

In addition to the feature space composed by density and density gradient magnitude, other features can be employed. Often, users who work on the multivariate time-varying data have specific domain knowledge regarding the properties and the distribution of the data. Expressing this distribution information with a set of GMM-enabled probabilistic representations in a selected feature space would best characterize the feature separation, and consequently facilitate a quick discovery of the particularly interesting features.

Fig. 9 displays the 30th out of 100 time steps of a vorticity field produced by an 128-cube simulation of a compressible and turbulent slip surface. Scientists assume that the vortex exists in the region of a large vorticity and a small pressure. The initial effort performed by our Volume Exploration tool produces 4 mixture components (top left) in the (vorticity, pressure) feature space, and discovers feature regions that contain too many vortices in the surface (middle). By examining the characterized regions of each ETF, the user finds that the cyan and yellow ETFs are the main cause of the ambiguity. Three operations are then performed, namely, remov-

ing these two ETFs, rescaling and recoloring the other two ETFs. The new result (see the right one in Fig. 9) reveals the kinking and tangling vortex tubes.

Note that, the feature space in this example does not contain arc-like shapes. It may be quite challenging for approaches that rely on the specification of the arc-shapes. In contrast, our approach can yield plausible results by leveraging the separation capability of the continuous and probabilistic representation with the GMM. Another advantage of the GMM is that it, by nature, takes a Gaussian form, and shares the benefits of the Gaussian Transfer Functions.

6.3 Discussion

Our volume exploration tool provides an easy-to-use exploration mechanism. However, automatically generating the maximized likelihood separation may lead to distracting noise or uninteresting materials, as shown in Fig. 7. Meanwhile, the current greedy EM algorithm does not allow for incorporating adjusted results into the clustering process. A new EM algorithm is needed to better utilize the users' prior knowledge in the optimization procedure.

Our current implementation only tests the maximized likelihood feature separation in the 2D feature spaces. It is straightforward to apply our scheme to a higher-dimensional feature space. The main challenge is that the performance will drop greatly with the increasing number of dimensions. In the future, we plan to develop a GPU implementation of an accelerated EM algorithm, and apply it to higher-dimensional feature spaces like the ones introduced in [20].

7 CONCLUSION

The paper introduces a new volume exploration scheme with the unique ability to capture the data characteristics, while still affording favorable user interactivity. This flexibility is especially helpful for inexperienced users because it can automatically provide a suggestive volume classification by means of a greedy EM algorithm. By allowing the user to interactively select pre-computed clusters in the feature space, he or she can have an initial picture of the underlying dataset. Next, each cluster can be manipulated with a suite of interactive widgets, whose consequences still take an ellipsoid gaussian form, i.e., they can be analytically integrated. By leveraging the power of GPGPU, the ETF-enabled transfer function can be seamlessly incorporated with the conventional pre-integrated volume rendering to generate high quality visualization results.

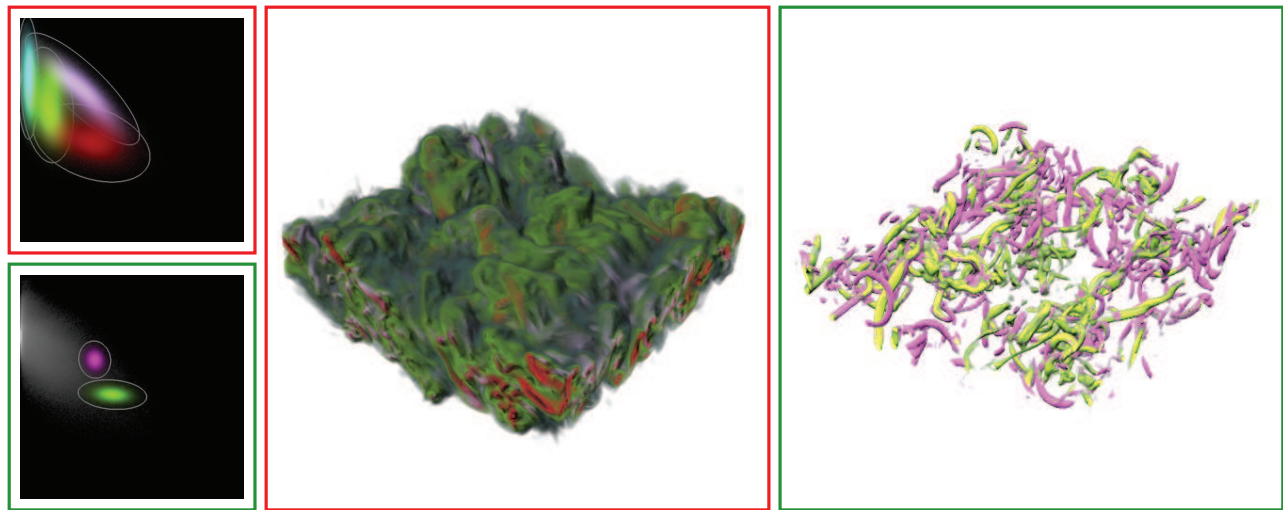


Figure 9: Exploring the Turbulent dataset in the (vorticity, pressure) feature space. The result (middle) with the automatically generated 4 ETFs (top left) does not clearly show the vortex tubes. By performing a sequence of operations, namely, removing the cyan and chartreuse ETFs, scaling the other two ETFs and recoloring them, the kinking and tangling vortex tubes are clearly shown with a large contrast.

ACKNOWLEDGEMENTS

We would like to thank Michael Knox for paper proofreading, Jun Liu for making figures and the anonymous reviewers for their valuable comments. This paper is partially supported by Knowledge Innovation Project of The Chinese Academy of Sciences (No.KGGX1-YW-13, No.O815011103), 973 program of China (2010CB732504), NSF of China (No.60873123, No.60873113), NSF of Zhejiang Province (No.1080618). The datasets are courtesy of General Electric, David Porter and Xinliang Li.

REFERENCES

- [1] J. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-02, University of Berkeley, 1998.
- [2] C. Correa, Y. Chan, and K. Ma. Uncertainty-Aware Visual Analytics. In *Proceedings of IEEE VAST 2009 Symposium (To Appear)*, 2009.
- [3] C. Correa and K. Ma. Size-based Transfer Functions: A New Volume Exploration Technique. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1380–1387, 2008.
- [4] C. Correa and K. Ma. The Occlusion Spectrum for Volume Visualization and Classification. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1465–1472, 2009.
- [5] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.
- [6] Y. Jang, R. Botchen, A. Lauser, D. Ebert, K. Gaither, and T. Ertl. Enhancing the interactive visualization of procedurally encoded multi-field data with ellipsoidal basis functions. *Computer Graphics Forum*, 25(3):587–596, 2006.
- [7] G. Kindlmann and J. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of IEEE Symposium on Volume Visualization '98*, pages 79–86, 1998.
- [8] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller. Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proceedings of IEEE Visualization '03*, pages 513–520, 2003.
- [9] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization '01*, pages 255–262, 2001.
- [10] J. Kniss, S. Premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization '03*, pages 65–72, 2003.
- [11] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.
- [12] R. Maciejewski, I. Wu, W. Chen, and D. Ebert. Structuring Feature Space: A Non-Parametric Method for Volumetric Transfer Function Generation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1473–1480, 2009.
- [13] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [14] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Avila, K. Martin, R. Machiraju, and J. Lee. The Transfer Function Bake-Off. *IEEE Computer Graphics and Applications*, 21(3):16–22, 2001.
- [15] S. Roettger, M. Bauer, and M. Stamminger. Spatialized transfer functions. In *Proceedings of IEEE/Eurographics Symposium on Visualization '05*, pages 271–278, 2005.
- [16] Y. Sato, C. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.
- [17] M. Selver and C. Güzelis. Semiautomatic Transfer Function Initialization for Abdominal Visualization Using Self-Generating Hierarchical Radial Basis Function Networks. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):395–409, 2009.
- [18] Y. Song, W. Chen, R. Maciejewski, K. Gaither, and D. Ebert. Bivariate Transfer Functions on Unstructured Grids. *Computer Graphics Forum*, 28(3):783–790, 2009.
- [19] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of CVPR '99*, pages 246–252, 1999.
- [20] F. Tzeng, E. Lum, and K. Ma. An Intelligent System Approach to Higher-Dimensional Classification of Volume Data. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):273–284, 2005.
- [21] F. Tzeng and K. Ma. A cluster-space visual interface for arbitrary dimensional classification of volume data. In *Proceedings of IEEE/Eurographics Symposium on Visualization '04*, pages 17–24, 2004.
- [22] J. Verbeek, J. Nunnink, and N. Vlassis. Accelerated EM-based clustering of large data sets. *Data Mining and Knowledge Discovery*, 13(3):291–307, 2006.
- [23] L. Wang, J. Giesen, K. McDonnell, P. Zolliker, and K. Mueller. Color Design for Illustrative Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1739–1754, 2008.